

Attorney Docket No. APL1P271/P3060

PATENT APPLICATION

NETWORK-BASED PURCHASE AND DISTRIBUTION OF MEDIA IN ACCORDANCE WITH PRIORITIES

Inventor: Jeffrey L. Robbin

Assignee: Apple Computer, Inc.

BEYER WEAVER & THOMAS, LLP
(650) 961-8300

NETWORK-BASED PURCHASE AND DISTRIBUTION OF MEDIA IN ACCORDANCE WITH PRIORITIES

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 60/465,410, filed April 25, 2003, and entitled "METHOD AND SYSTEM FOR SECURE NETWORK-BASED DISTRIBUTION OF MEDIA", and which is hereby incorporated by reference herein.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present invention relates to performance of media-based operations on a computer and, more particularly, to prioritization of media-based operations to be performed on a computer.

Description of the Related Art

[0003] Traditionally, music has been purchased at music stores or music departments of larger stores. A consumer will visit the music store or department and manually browse for albums or compact discs (CDs) of interest. Often, the music in the music store or department is categorized by genre, and then indexed by artist. For example, genre can include rock, country, pop, soul, jazz, etc. After the consumer selects an album or CD of interest, the consumer proceeds to a check-out register to pay for the album or CD being purchased.

[0004] In recent years music delivery or distribution over the Internet has become popular. Due to the advances in efficient file formats, such as MP3 and MPEG4, the size of media files have become small enough to make their download via the Internet practical. Also, technological advances have led to higher-speed Internet connections and lower cost of memory. The combination of these advances make downloading media files, such as for music and videos, manageable and not too time consuming. Normally, client-side programs assist a user with interaction via the Internet with a server computer that stores available media files. In this way, a user

of a client-side program is able to browse, preview, purchase and/or download available media files over the Internet. These operations can be referred to as media operations.

[0005] Given that media operations are bandwidth-intensive, a user may desire to perform several media operations concurrently. Unfortunately, however, the user can experience delays if earlier media operations are slow to complete. For example, the download of a media file to a client can take at least several minutes, and during this period, the user may want to perform another media action (e.g., browse). However, the subsequent media operation must await the completion of the download operation. Delays of this sort are bothersome to users and tend to reduce user satisfaction.

[0006] Thus, there is a need for media operations to be processed in a manner that enhances user satisfaction.

SUMMARY OF THE INVENTION

[0007] Broadly speaking, the invention relates to prioritizing use of resources (e.g., network resources) at a client machine (e.g., computer) so that user satisfaction is enhanced. The client machine typically executes a software program that permits a user of the client machine to browse, preview, purchase, or download media. The media can, for example, be audio, video, or image data. More particularly, in the case of a client machine that operates at least one application program that provides media-based actions, the invention manages the use of limited network resources (e.g., network connection bandwidth) at the client machine so that different media-based actions are performed at different priority levels. For example, at the client machine, a second type of media-based action can be issued while a first type of media-based action is being processed. When the second type of media-based action has a higher priority level than the first type of media-based action, then the processing of the first type of media-based action can be suspended so that processing of the second type of media-based action can be given preference.

[0008] The invention can be implemented in numerous ways, including as a method, system, device, apparatus, graphical user interface, or computer readable medium. Several embodiments of the invention are discussed below.

[0009] As a method for managing tasks performed on a computer capable of coupling over a network to a network-based media server, one embodiment of the invention includes at least the acts of: receiving tasks to be performed from at least one client application operating on the computer, the tasks pertaining to one or more different media-based actions, and the tasks involving interaction with the media server over the network; activating an operation at the computer to respond to each of the tasks; and coordinating performance of the activated operations in accordance with priority levels associated with the different media-based actions of the tasks, the different ones of the different media-based actions having different priority levels.

[0010] As a computer readable medium including at least computer program code for managing tasks performed on a computer capable of coupling over a network to a network-based media server, one embodiment of the invention includes at least: computer program code for receiving tasks to be performed from at least one client application operating on the computer, the tasks pertain to one or more different media-based actions, and the tasks involving interaction with the media server over the network; and computer program code for coordinating performance of the tasks in accordance with priority levels associated with the different media-based actions of the tasks, the different ones of the different media-based actions having different priority levels.

[0011] As a computer for presenting media to its user, one embodiment of the invention includes at least: a client media player program operable to enable the user to play, browse, preview, purchase or download present media items for the benefit of the user; a network interface that permits the client media application program to interact with a media commerce server that stores or manages a plurality of media items that can be browsed, previewed, purchased or downloaded; and a task manager that manages performance of at least browse, preview, purchase or download operations by assigning priority levels to each of the browse, preview, purchase or download operations, and managing performance of the browse, preview, purchase or download operations in accordance with the assigned priority levels.

[0012] Other aspects and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying drawings which illustrate, by way of example, the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein like reference numerals designate like structural elements, and in which:

[0014] FIG. 1 is a flow diagram of an execution management process according to one embodiment of the invention.

[0015] FIGs. 2A and 2B are flow diagrams of an operation scheduling process according to one embodiment of the invention.

[0016] FIG. 3 is a flow diagram of a priority update process according to one embodiment of the invention.

[0017] FIG. 4 is a diagram depicting an exemplary thread execution according to one embodiment of the invention.

[0018] FIG. 5 is a diagram of a software stack.

[0019] FIG. 6 is a task priority scheduling system according to one embodiment of the invention.

[0020] FIG. 7 is a block diagram of a media purchase system according to one embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0021] The invention relates to prioritizing use of resources (e.g., network resources) at a client machine (e.g., computer) so that user satisfaction is enhanced. The client machine typically executes a software program that permits a user of the client machine to browse, preview, purchase, or download media. The media can, for example, be audio, video, or image data. More particularly, in the case of a client machine that operates at least one application program that provides media-based actions, the invention manages the use of limited network resources (e.g., network

connection bandwidth) at the client machine so that different media-based actions are performed at different priority levels. For example, at the client machine, a second type of media-based action can be issued while a first type of media-based action is being processed. When the second type of media-based action has a higher priority level than the first type of media-based action, then the processing of the first type of media-based action can be suspended so that processing of the second type of media-based action can be given preference.

[0022] Embodiments of the invention are discussed below with reference to FIGs. 1 – 7. However, those skilled in the art will readily appreciate that the detailed description given herein with respect to these figures is for explanatory purposes as the invention extends beyond these limited embodiments.

[0023] FIG. 1 is a flow diagram of an execution management process 100 according to one embodiment of the invention. The execution management process 100 can be part of a client program operating on a client machine. The client program can pertain to an application program or an operating system program.

[0024] The execution management process 100 initially receives 102 tasks to be executed on a client machine (client computer). The tasks pertain to different media-based actions that can be carried out on the client machine. The different media-based actions are often induced by a user that interacts with the client machine. Typically, the client machine couples to a server computer through a network (such as the Internet). The different media-based actions often require interaction with the server computer.

[0025] After the tasks for the different media-based actions have been received 102, an operation is activated 104 to respond to each of the tasks. In other words, for each task that is received, an operation is activated 104. Since the client machine can operate in a multi-threaded fashion, it is able to support processing of a large number of operations. In one implementation, an operation is carried out at the client machine by one or more threads. Nevertheless, the operations need to be managed such that the responsiveness of the client machine to the various tasks is appropriate. In one embodiment, a user can set the priority levels for different media-based actions in accordance with their desires. In another embodiment, the

client machine or client program can provide predetermined priorities to the various different media-based actions.

[0026] In any case, after the operations have been activated 104, the execution management process 100 coordinates 106 performance of the activated operations in accordance with priority levels associated with the different media-based actions. In this regard, certain of the operations can be suspended while other operations are performed so that the priority levels for the different media-based actions can be enforced. Once all the operations have been completed, the execution management process 100 is complete and ends. However, it should be noted that the execution management process 100 could also return to repeat the operation 102 and subsequent operations so that subsequently received tasks for different media-based actions can be similarly processed.

[0027] FIGs. 2A and 2B are flow diagrams of an operation scheduling process 200 according to one embodiment of the invention. The operation scheduling process 200 is, for example, performed on a client computer. The operation scheduling process 200 begins with a decision 202 that determines whether a new media task has been received. When the decision 202 determines that a new media task has been received, the operation scheduling process 200 continues. In other words, the operation scheduling process 200 can be deemed invoked whenever a new media task is received. In one embodiment, the tasks pertain to media-based actions. Examples of media-based actions include at least: previewing music, browsing music, purchasing music, and downloading music. These exemplary media-based actions normally would require the client computer to interact over a network with a server computer, which serves as a media server. Downloading is the most bandwidth intensive of these exemplary media-based actions, though previewing a media item can involve streaming a small segment of the media file and can thus also be bandwidth-intensive. Purchasing does not require too much bandwidth but can be time-intensive for authorization of payments, etc. Browsing may be bandwidth-intensive if the webpage presented to the client computer includes complex graphics.

[0028] Once the decision 202 determines that a new media task has been received, the type of media task is identified 204. Then, the priority for the type of media task is determined 206. In one implementation, a table can store a priority

indication for each of a plurality of different types of media tasks. Such a table can be associated with the particular client machine, a single user, or a plurality of users. After the priority for the type of media task has been determined 206, an operation for a response to the media task is opened 208. Then, a priority is assigned 210 to the operation opened based on the determined priority.

[0029] At this point, the operation can proceed to execute in order to produce the response to the media task. However, its performance is dependent upon other performing other operations within the environment of the client computer (and perhaps also responsiveness of the server computer). In one implementation, the environment of the client computer is a multi-threaded environment. A decision 212 determines whether any existing operations that are activated have lower priorities. When the decision 212 determines that there are existing operation with lower priorities, then the one or more existing operation that have a lower priority are suspended 214. When the decision 212 determines that there are no existing operations with lower priorities, the operation 214 can be bypassed.

[0030] Following the operation 214 or its being bypassed, a decision 216 determines whether the task has been completed. The task can, for example, be considered completed when the corresponding operation or operations complete or when a response (or results) of the task have been presented at the client computer. When the decision 216 determines that the task has not completed, the operation scheduling process 200 awaits the completion of the task. Once the decision 216 determines that the task has been completed, the operation that produced the response is closed 218. Thereafter, the one or more existing operations (if any) that were suspended are resumed 220. Following the operation 220, the operation scheduling process 200 is complete and ends.

[0031] FIG. 3 is a flow diagram of a priority update process 300 according to one embodiment of the invention. The priority update process 300 is a process performed at a client computer in response to a request, namely, a priority setting request from a client program or client machine. The priority update process 300 begins with a decision 302 that determines whether a priority setting request has been received. The priority setting request could be initiated by the client program itself (i.e., automatically) or by a user interacting with the client program or client machine (i.e., user-initiated). When the decision 302 determines that a priority

setting request has not been received, then the priority update process 300 awaits such a request. Once the decision 302 determines that a priority setting request has been received, then the priority levels for the user or the client program are updated 304. Here, the priority setting request can contain information that specifies one or more priority levels to be utilized. To the extent priority levels were previously established, these newly provided priority levels operate to update the previous priority levels. In any case, after the priority levels have been updated 304, the updated priority levels are stored 306. Here, the updated priority levels are stored 306 to the client computer. Following the operation 306, the priority update process 300 is complete and ends.

[0032] As noted above, for each task to be performed, an operation can be invoked. In one embodiment, the computer system provides a multi-threaded environment and each operation can be implemented as one or more threads.

[0033] FIG. 4 is a diagram depicting an exemplary thread execution 400 according to one embodiment of the invention. The exemplary thread execution 400 illustrates operation of a series of threads in response to a series of tasks to be performed. At time T_0 task A is requested at a client computer. The client computer opens thread A to satisfy task A. At time T_0+x , while the thread A is executing, the client computer receives task B. The client computer then opens thread B to process the task B. Additionally, it is determined at the client computer that the thread B is a higher priority thread than thread A. Consequently, thread A is suspended, thereby allowing thread B to execute with greater performance.

[0034] According to the exemplary thread execution, the thread B completes while thread A is suspended. Then, thread B is closed. At this point, the higher priority thread B is no longer present, so the thread A is resumed from its suspended state. The thread A then continues to execute. However, at time T_0+y , the client computer receives task C. The client computer then opens thread C to process the task C. Again, it is determined that thread C is the higher priority than thread A. As a result, thread A is again suspended. Then, thread C executes to its completion. Thereafter, thread C closes, and thread A resumes. Subsequently, thread A completes and thread A is then closed as its associated task has fully completed.

[0035] The tasks A, B and C presented in the exemplary thread execution 400 illustrated in FIG. 4 can pertain to the same or different tasks. Typically, the tasks are of different types and thus have different priority levels. These priority levels are utilized in determining which of the activated threads are to be suspended so as to give priority of execution to other threads. Still further, it should be understood that the tasks are received at arbitrary times at the client computer and thus various combinations or interactions of active threads can occur at the client computer. The priority levels are utilized so as to provide the user and/or client application with the opportunity to control the responsiveness of the client computer to certain types of requests versus other types of requests.

[0036] FIG. 5 is a diagram of a software stack 500. The software stack 500 is a basic representative stack of software components that would be provided at a client computer. The software stack 500 includes a network interface 502 that couples to the physical network, an operating system 504, and an application 506. Typically, there are multiple applications 506 on the software stack 500. The task execution management associated with the invention can be implemented at the application 506 or, alternatively, at the operating system 504. Typically, the operating system 504 supports multi-threaded operations, and the application is 506 multi-threaded or multi-threaded safe. The advantage of implementing the task execution management at the operating system 504 is that the management would be able to be cross-application, whereas when implemented at the application 506, the management would only be intra-application.

[0037] FIG. 6 is a task priority scheduling system 600 according to one embodiment of the invention. The task priority scheduling system 600 is, for example, implemented at a client computer. The task priority scheduling system 600 includes a task manager 602. The task manager 602 operates to schedule which of the task that are to be performed based on priority levels. The task manager 602 can activate various different types of media-based actions, through activation of operations (e.g., threads) at the client computer. Namely, the task manager can activate a browse task 604, a preview task 606, a purchase task 608 and/or a download task 610. For example, the client computer might receive a task for a browse operation, and thus the task manager 602 would activate the browse task 604 instance for processing the browse operation. Similarly, if the client computer

were to receive a task for a preview operation, then the task manager 602 would activate a preview task 606 instance so that a preview media item (e.g., small audio file) could be received by the requestor. The task manager 602 would interact with a priorities table 612 to assist with the management of the execution of the multiple tasks that are typically operating. In this regard, each of the different types of tasks would be provided with a priority level in the priorities table 612. The task manager 602 could then read the priority level for the associated type of action and cause other lower priority actions to be suspended as appropriate. As a result, the system provides greater execution performance (and thus greater responsiveness) to higher priority actions.

[0038] One example of a priorities table suitable for used as the priorities table 612 illustrated in FIG. 6 is as follows:

Media-based action	Priority
Browse	8
Preview	10
Purchase	6
Download	4

where "Browse," "Preview," "Purchase," and "Download" are exemplary media-based actions that can be processed by operations. In this example, the priority ratings for these actions are be "8," "10," "6," and "4," respectively. These priority ratings are provided on a 1-10 scale, with "10" being highest priority. Various other rating schemes can be used so long as the relative priority of the different actions can be represented.

[0039] The invention can be used in conjunction with a system and method for purchasing media items over a network. A potential purchaser (i.e., user) can search and browse through numerous media items that are available for purchase. A potential purchaser can then purchase a media item with great ease. Upon purchasing a media item, the content for the media item can be downloaded over the

network to the purchaser. The content for the media item is then encrypted for the purchaser's use and stored on the purchaser's machine. Thereafter, the purchaser can make use of the media item (e.g., play the media item). However, the use of the media item can still be limited. For example, only up to a predetermined number user machines can be authorized to use the media item, or only up to a predetermined number of compact disc copies can be made of a group or collection of media items (e.g., a playlist).

[0040] FIG. 7 is a block diagram of a media purchase system 700 according to one embodiment of the invention. The media purchase system 700 includes a media commerce server 702. The media commerce server 702 coordinates review and/or purchase of media items through on-line transactions. On-line transactions to purchase media items is also referred to as electronic commerce (e-commerce). The media purchase system 700 also includes a client 704. Typically, the media purchase system 700 would include a plurality of different clients 704. Each client 704 includes a media player 708. The media player 708 is an application program (e.g., software application) that operates on the client 704, which is a computing device. The client 704 is coupled to the media commerce server 702 through a data network 706. Hence, any of the clients 704 can interact with the media commerce server 702 to review and/or purchase media items. In one embodiment, the data network 706 includes at least a portion of the Internet. The clients 704 can vary with application but generally are computing devices that have memory storage. Often, the clients 704 are personal computers or other computing devices that are capable of storing and presenting media to their users.

[0041] The media purchase system 700 also includes a media storage server 710 and a media store 712. The media storage server 710 represents a remote storage server that couples to the data network 706. The media store 712 provides mass storage of the numerous media items that are available for purchase via the media purchase system 700. Once purchased, the media items can be accessed from the media store 712 over the data network 706 by way of the media storage server 710.

[0042] More particularly, the media purchase system 700 allows a user of the client 704 to utilize the media player 708 to browse, search or sort through a plurality of media items that can be purchased from the media commerce server 702. The media player 708 may also allow the user to preview a media clip of the media items.

In the event that the user of the media player 708 desires to purchase a particular media item, the user (via the media player 708) and the media commerce server 702 engage in an on-line commerce transaction in which the user pays for access rights to the particular media item. In one embodiment, a credit card associated with the user is credited for the purchase amount of the particular media item.

[0043] In the media purchase system 700 shown in FIG. 7, the media items are stored in the media store 712 and retrieved via the media storage server 710. Hence, the media commerce server 702 need not burden its resources to deliver any of the media items that may be purchased to the client 704. Instead, on purchasing a particular media item, the media commerce server 702 sends download information to the media player 708 on the client 704. The download information can then be used by the media player 708 (and the client 704) to retrieve the media content for the particular media item by interacting with the media storage server 710 through the data network 706. In this regard, the media storage server 710 obtains the media content corresponding to the particular media item from the media store 712 and downloads such content through the data network 706 to the client 704. The downloaded media content can then be stored on the client 704. In one embodiment, the downloaded media content is encrypted as received at the client 704 but is decrypted and then re-encrypted before persistent storage on the client 704. Thereafter, the media player 708 can present (e.g., play) the media content at the client 704.

[0044] The connections through the data network 706 between the media commerce server 702, the client 704 and the media storage server 710 can be through secure connections, such as Secure Sockets Layer (SSL). Further, the media content is re-encrypted prior to storage at the client 704 such that downloaded media content is not stored in the clear, but is instead stored in an encrypted manner.

[0045] The media items can pertain to audio items (e.g., audio files or songs, such as for music), video items (e.g., video files or movies), or image items (e.g., photos).

[0046] The invention is preferably implemented by software, but can also be implemented in hardware or a combination of hardware and software. The invention

can also be embodied as computer readable code on a computer readable medium. The computer readable medium is any data storage device that can store data which can thereafter be read by a computer system. Examples of the computer readable medium include read-only memory, random-access memory, CD-ROMs, DVDs, magnetic tape, optical data storage devices, and carrier waves. The computer readable medium can also be distributed over network-coupled computer systems so that the computer readable code is stored and executed in a distributed fashion.

[0047] The advantages of the invention are numerous. Different embodiments or implementations may, but need not, yield one or more of the following advantages. One advantage of the invention user satisfaction is improved by making client-side media programs more responsive to users. Another advantage of the invention is that software developers can control responsiveness of client-side media programs based on different media-based actions to be performed. Still another advantage of the invention is that network bandwidth is able to be utilized in a manner that renders client-side media programs more responsive or user friendly.

[0048] The many features and advantages of the present invention are apparent from the written description and, thus, it is intended by the appended claims to cover all such features and advantages of the invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, the invention should not be limited to the exact construction and operation as illustrated and described. Hence, all suitable modifications and equivalents may be resorted to as falling within the scope of the invention.

What is claimed is: